Ruler of the Plane – Games of Geometry

Sander Beekhuis[1], Kevin Buchin[2], Thom Castermans[3], Thom Hurks[4], and Willem Sonke[5]

1 **Eindhoven University of Technology, Eindhoven, The Netherlands**
2 **Eindhoven University of Technology, Eindhoven, The Netherlands**
`k.a.buchin@tue.nl`
3 **Eindhoven University of Technology, Eindhoven, The Netherlands**
`t.h.a.castermans@tue.nl`
4 **Eindhoven University of Technology, Eindhoven, The Netherlands**
5 **Eindhoven University of Technology, Eindhoven, The Netherlands**
`w.m.sonke@tue.nl`

Abstract

Ruler of the Plane is a set of games illustrating concepts from combinatorial and computational geometry. The games are based on the art gallery problem, ham-sandwich cuts, the Voronoi game, and geometric network connectivity problems like the Euclidean minimum spanning tree and traveling salesperson problem.

1998 ACM Subject Classification F.2.2 [Nonnumerical Algorithms and Problems] Geometrical Problems and Computations

Keywords and phrases art gallery problem, ham-sandwich cuts, Voronoi game, traveling salesperson problem

Digital Object Identifier 10.4230/LIPIcs.SoCG.2017.63

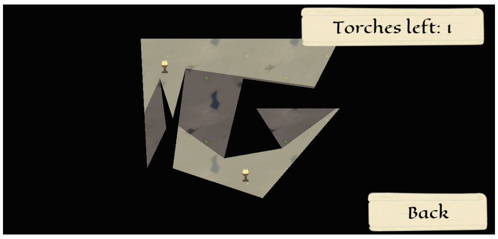
Category Multimedia Contribution

1 Concept

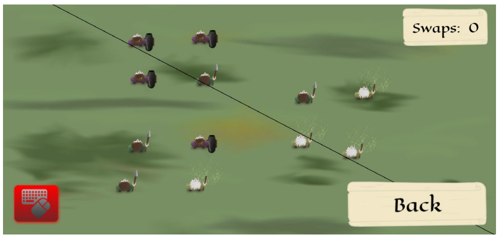
Geometry being inherently tangible, lends itself to be the base of puzzles and games. *Ruler of the Plane* is a set of four games with a medieval theme illustrating concepts from combinatorial and computational geometry. The games are based on the art gallery problem, ham-sandwich cuts, the Voronoi game, and geometric network connectivity problems like the Euclidean minimum spanning tree and traveling salesperson problem (TSP), see Figure 1.

The games also aim at providing the interested player with background on the geometric algorithms and data structures needed to implement such games. They do so by providing some pointers to geometric concepts in the game explanations, and by allowing to visualize some of the underlying data structures. For instance, the game on the ham-sandwich cuts can show the dual arrangements of the different color classes, the Voronoi game allows to show the Delaunay triangulation and empty circles. Furthermore, the games are open source and implemented using *C#* in the game engine Unity, and therefore provide the possibility to explore the underlying algorithms and data structures.

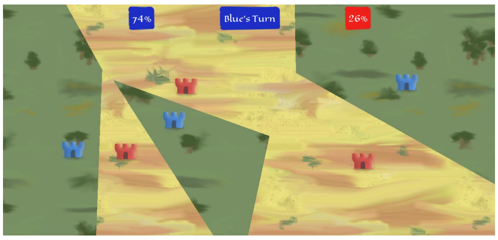
The geometric problems and the underlying algorithms and data structures of the games are common content of a Computational Geometry course. We developed the game primarily to introduce students taking such a course to these concepts in an entertaining way. An additional goal is to provide a stepping stone to introduce Combinatorial and Computational Geometry and also other algorithmic concepts like NP-hardness problems to a wider audience.



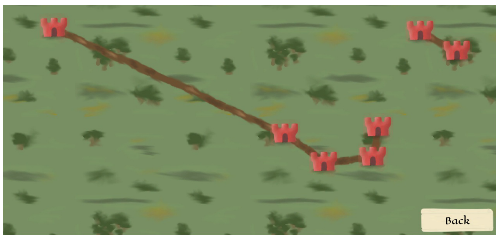
(a) *illuminate* a dungeon



(b) *divide* forces into equal units



(c) *conquer* new lands



(d) *connect* your new settlements

Figure 1 Games in *Ruler of the Plane*.

2 The Games

The game *conquer* implements the classical Voronoi game [1]: Two players place castles in turn, and the player whose Voronoi regions occupy the most area at the end wins.

The Voronoi diagram is implemented as dual of the Delaunay triangulation. The Delaunay triangulation is constructed using an implementation of a textbook randomized incremental construction [6]. Out of the four games, this is the only two-player game. To demonstrate the underlying geometry the game allows to toggle the Voronoi diagram, empty circumcircles and the dual Delaunay triangulation (see Figure 2a).

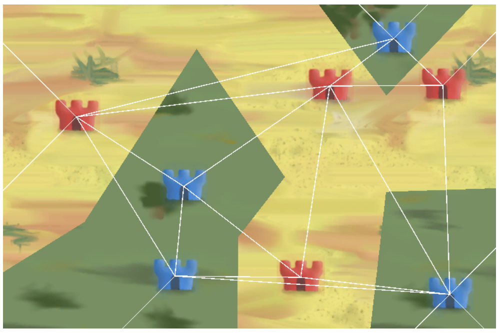
The game *divide* implements the two-dimensional ham sandwich cuts [9], but with three types of points. That is, the player needs to find a line that splits all three types of points in half. Some levels also ask to swap positions of points, before drawing a cut.

In a course on computational geometry, ham sandwich cuts are commonly covered as an application of duality and arrangements. In this context typically a simple $O(n^2)$-time algorithm is discussed: dualizing the points, computing the line arrangements and intersecting the $n/2$ levels. This is also the algorithm implemented in the game. The game allows to toggle possible cuts and the dual arrangements (see Figure 2b).

The game *connect* consists of three separate games. In the first the player has to find the Euclidean minimum spanning tree, in the second a Euclidean traveling salesperson tour, and in the third a 1.5-spanner [11] of short length. While in the first game the player has to find the exact tree, in the two other games the player has to beat an approximation computed by the game, namely Christofides algorithm [7] and the greedy spanner [4].

We included the TSP with Christofides algorithm and minimum spanning trees, since they are very natural geometric problems, suitable to discuss computational complexity with a wider audience, and since they often feature in other algorithms courses. spanners are often discussed in the context of well-separated pair decompositions.

The spanner game also provides a limited number of 'hints' in the form of the next edge the greedy spanner would add. After exhausting the base levels, the game continues with levels that ask to connect randomly generated sites.



(a) Delaunay triangulation of the castles

(b) cells in the dual arrangement

Figure 2 Visualizing the underlying algorithms and data structures.

The game *illuminate* is an implementation of art gallery problem [2] with point guards in a simple polygon. In a Computational Geometry course, the art gallery problem with vertex guards is often discussed as a motivation for polygon triangulation, but is also interlinked with other topics, like visibility computation and boolean operations on polygons. The game computes visibility regions by a circular sweep. To remove duplicate regions it then uses the Weiler–Atherton algorithm [13]. The implementation is not yet robust, and therefore only small levels are included in the game.

3 Educational Context

As described above the games are intended for demonstration purposes for students of Computational Geometry and for a wider audience. However, also the game development was embedded in an educational context.

Various concepts for games where first implemented and tested as course projects in Computational Geometry. Some of these concepts where then integrated into the game. Most of *Ruler of the Plane* was then implemented by Master students after taking a course in Computational Geometry, partially as practical component to a reading course on algorithm engineering [10] and robust geometric algorithms [12], partially as student assistantship. The task to extend the games may result in engaging future course projects. Ruler of the Plane is open-source using *C#* in the game engine Unity, and therefore lends itself to such extensions.

4 Future Work

The games leave many opportunities for future work from designing interesting levels and variants, to improving and providing alternative implementations, to designing games on other geometric and algorithmic topics. In the following we discuss some more concrete ideas.

The Voronoi game allows for a very simple, effective strategy, which in the basic variant makes the game *conquer* less interesting. Including other variants would make the game more multi-facetted. These could include castles with different magnitudes/ranges of influence, or region of different worth, or restrictions on where castles can be placed. Also a puzzle variant where castles of one color are already placed, and the player only places castles of the other color could be challenging.

Currently the game *divide* has a small number of levels. It would be possible to generate additional levels based on random instances, but the question of generating challenging levels remains open. And again, more variants could bring more variation to the game.

In the game *connect* Christofides algorithm may be instructive, but the results are quite easy to beat. To demonstrate the NP-hardness of the problem, it would be interesting to include small, difficult instances. Other TSP heuristics would be instructive as 'hints'.

Generating interesting levels for the art gallery problem in the game *illuminate* seems challenging. Possible starting points could be gadgets used in NP-hardness constructions [8] and families of polygons used in experimental evaluations [5]. The art gallery problem has many variants, and more of these would again bring more variation to the game. In particular vertex guards would provide a game more closely to the art gallery problem as motivation to polygon triangulation.

So far games about four topics have been implemented. Obviously a course on Computational Geometry [6] leaves room for more games on other topics. For some topics, for instance orthogonal range searching, it might be more challenging to design an interesting game. And then there are other topics, which are intuitively accessible and seem to lend themselves as a base of a game, e.g, man-and-dog problems [3].

5 Resources

The games can be played online at `http://www.win.tue.nl/~kbuchin/proj/ruler/webgl/`. The game can be downloaded from `http://www.win.tue.nl/~kbuchin/proj/ruler/` and its sources from `https://github.com/kbuchin/ruler`. A video demonstrating the game is available at `http://www.win.tue.nl/~kbuchin/proj/ruler/video/PlaneRuler.mp4`.

Acknowledgments. The development of the game was supported by the TU/e Educational award.

References

1 Hee-Kap Ahn, Siu-Wing Cheng, Otfried Cheong, Mordecai Golin, and Rene Van Oostrum. Competitive facility location: the Voronoi game. *Theoretical Computer Science*, 310(1-3):457–467, 2004.
2 Martin Aigner and Günter M. Ziegler. *Proofs from THE BOOK*. Springer, 4th edition, 2009.
3 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.*, 5:75–91, 1995.
4 Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993.
5 Yoav Amit, Joseph S. B. Mitchell, and Eli Packer. Locating guards for visibility coverage of polygons. *Int. J. Comput. Geom. Appl.*, 20:601–630, 2010.
6 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2008.
7 Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
8 Der-Tsai Lee and Arthur K. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986.
9 Chi-Yuan Lo, Jiří Matoušek, and William Steiger. Algorithms for ham-sandwich cuts. *Discrete & Computational Geometry*, 11(4):433–452, 1994.
10 Matthias Müller-Hannemann and Stefan Schirra, editors. *Algorithm engineering: bridging the gap between algorithm theory and practice*, volume 5971 of *LNCS*. Springer, 2010.

11 Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, New York, NY, USA, 2007.

12 Jonathan Richard Shewchuk. Lecture notes on geometric robustness, 2013.

13 Kevin Weiler and Peter Atherton. Hidden surface removal using polygon area sorting. *ACM SIGGRAPH computer graphics*, 11(2):214–222, 1977.